

WEST[Help](#)[Logout](#)[Interrupt](#)[Main Menu](#)[Search Form](#)[Posting Counts](#)[Show S Numbers](#)[Edit S Numbers](#)[Preferences](#)[Cases](#)**Search Results -**

Term	Documents
VIRTUAL	55916
VIRTUALS	3
SERVER\$	0
SERVER	36819
SERVERA	9
SERVERABILITY	1
SERVERABLE	11
SERVERABLY	2
SERVERACCEPT	1
SERVERACCESS	1
"SERVERACCESS.JAVA"	1
(L1 AND (VIRTUAL ADJ1 SERVER\$).AB.).USPT.	5

[There are more results than shown above. Click here to view the entire set.](#)

Database:

US Patents Full-Text Database
US Pre-Grant Publication Full-Text Database
JPO Abstracts Database
EPO Abstracts Database
Derwent World Patents Index
IBM Technical Disclosure Bulletins

Search:

L2

[Refine Search](#)[Recall Text](#)[Clear](#)**Search History**

DATE: Monday, July 07, 2003 [Printable Copy](#) [Create Case](#)

Set Name Query
side by side

Hit Count Set Name
result set

DB=USPT; PLUR=YES; OP=ADJ

L2 L1 and (virtual adj1 server\$.ab. 5 L2

L1 ((709/\$)!.CCLS.) 16186 L1

END OF SEARCH HISTORY

WEST[Generate Collection](#)[Print](#)**Search Results - Record(s) 1 through 5 of 5 returned.**☐ 1. Document ID: US 6286047 B1

L2: Entry 1 of 5

File: USPT

Sep 4, 2001

DOCUMENT-IDENTIFIER: US 6286047 B1

TITLE: Method and system for automatic discovery of network services

Abstract Text (1):

A method for identifying services, service elements and dependencies among the services and service elements includes executing first and second phases of discovery. In the first phase, the services and service elements are detected, as well as a first set of dependencies. The second phase is based on results of the first phase and is focused upon detecting inter-service dependencies, i.e., conditions in which proper operation of one service relies upon at least one other service. Various techniques may be used in executing the first phase, including accessing information in a domain name service (DNS) of the network to identify dependencies, as well as services and service elements. Discovery within the first phase may also be based upon recognizing naming conventions. Regarding the second phase, one approach to discovering inter-service dependencies is to deploy discovery agents implemented in computer software to access content of configuration files of applications detected in the first phase. Discovery agents may also be used to monitor connections completed via specified service elements detected in the first phase, such that other inter-service dependencies are identified. As an alternative or additional approach, network probes may be deployed to access information of data packets transmitted between service elements detected in the first phase, with the accessed packet information being used to detect inter-service dependencies. When information of the DNS is accessed in the first phase, the information is used as a basis for determining at least some of (1) groups of service elements that are generally equivalent with respect to executing a particular service within the network, (2) hosts supporting virtual hosting, (3) hosts supporting virtual servers, and (4) name servers.

Current US Original Classification (1):709/224Current US Cross Reference Classification (5):709/202Current US Cross Reference Classification (6):709/217Current US Cross Reference Classification (7):709/226

Full	Title	Citation	Front	Review	Classification	Date	Reference	Sequences	Attachments	Claims	KWIC	Draw Desc	Image
------	-------	----------	-------	--------	----------------	------	-----------	-----------	-------------	--------	------	-----------	-------

☐ 2. Document ID: US 6092178 A

L2: Entry 2 of 5

File: USPT

Jul 18, 2000

DOCUMENT-IDENTIFIER: US 6092178 A

**** See image for Certificate of Correction ****

TITLE: System for responding to a resource request

Abstract Text (1):

A trigger is provided in association with a network naming service, such as DNS (Domain Name Service), that handles client requests for an application. The trigger comprises a set of executable instructions referenced by a resource record associated with an identifier of the application. In response to a client request concerning the application, the resource record is retrieved and the instructions are executed. In one implementation of a trigger, a DNS server provides load balancing among a plurality of servers within a network name space (e.g., domain or sub-domain) offering an application program (or replicated service) that is known by a virtual server name. A policy is selected for choosing a preferred server from the plurality of servers according to a specified status or operational characteristic of the application instances, such as the least-loaded instance of the application or the instance with the fastest response time. The policy is encapsulated within multiple levels of objects or modules distributed among the plurality of servers and the DNS server. The objects collect and assemble the servers' status and operational characteristics. The information collected by the objects is analyzed to select the server that best satisfies the selected policy. A client request for the application is received by the DNS server, which retrieves a resource record corresponding to the virtual server name. Within the record is the name of a trigger. The trigger is executed to select, or retrieve an identity of, a server to which the client request is to be directed.

Current US Cross Reference Classification (1):709/105

Full	Title	Citation	Front	Review	Classification	Date	Reference	Sequences	Attachments	Claims	KWIC	Draw Desc	Image
------	-------	----------	-------	--------	----------------	------	-----------	-----------	-------------	--------	------	-----------	-------

☐ 3. Document ID: US 5870550 A

L2: Entry 3 of 5

File: USPT

Feb 9, 1999

DOCUMENT-IDENTIFIER: US 5870550 A

TITLE: Web server employing multi-homed, modular framework

Abstract Text (1):

A Web server is provided having a multi-homed, modular framework. The modular framework allows extensions to the Web server to be easily compiled into the Web server, allowing the extensions to run natively as part of the server instead of incurring the overhead typical of CGI scripts, for example. The multi-homing capabilities of the Web server provide the appearance to Web users of multiple distinct and independent servers, allowing a small company or individual to create the same kind of Web presence enjoyed by larger companies. In effect, multiple virtual servers run on the same physical machine. The Web server as a whole is easily extensible to allow additional capabilities to be provided natively within the Web server itself. Furthermore, each virtual server is independently configurable in order to turn different capabilities on or off or to modify operation of the virtual server. The Web server is also provided with enhanced security features, built-in animation capability, and other features that afford maximum flexibility and versatility.

Current US Original Classification (1):709/218

Current US Cross Reference Classification (2):
709/219

Full	Title	Citation	Front	Review	Classification	Date	Reference	Sequences	Attachments
------	-------	----------	-------	--------	----------------	------	-----------	-----------	-------------

KWAC	Draw Desc	Image
------	-----------	-------

☐ 4. Document ID: US 5602992 A

L2: Entry 4 of 5

File: USPT

Feb 11, 1997

DOCUMENT-IDENTIFIER: US 5602992 A

TITLE: System for synchronizing data stream transferred from server to client by initializing clock when first packet is received and comparing packet time information with clock

Abstract Text (1):

An apparatus and method for synchronizing multiple independent data streams in a networked computer system is disclosed. In a computer network having a plurality of computer systems coupled by a communication medium, one of the plurality of computer systems being a server system, a different one of the plurality of computer systems being a client system, a method for synchronizing a data stream transferred from the server system to the client system, the method comprising the steps of: 1) capturing information into a data packet, the capturing step being performed by the server system; 2) storing time information into the data packet, the time information indicating when the capturing step is complete; 3) transferring the data packet to the client system across the communication medium; 4) initializing a virtual server system clock if the data packet is a first data packet is the data stream; 5) retrieving the time information from the data packet; 6) comparing the time information with the virtual server system clock; and 7) processing the information in the data packet if the time information indicates a time equal to a time indicated by the virtual server system clock. The present invention further includes the steps of: 1) scheduling processing of the information in the data packet if the time information indicates a time later than a time indicated by the virtual server system clock; and 2) discarding the data packet if the time information indicates a time earlier than a time indicated by the virtual server system clock.

Current US Original Classification (1):
709/248

Full	Title	Citation	Front	Review	Classification	Date	Reference	Sequences	Attachments
------	-------	----------	-------	--------	----------------	------	-----------	-----------	-------------

KWAC	Draw Desc	Image
------	-----------	-------

☐ 5. Document ID: US 5450583 A

L2: Entry 5 of 5

File: USPT

Sep 12, 1995

DOCUMENT-IDENTIFIER: US 5450583 A

**** See image for Certificate of Correction ****

TITLE: Object-oriented language processing system

Abstract Text (1):

A method for realizing an object-oriented language interface. A server process is performed for operating an object written in a first object-oriented language, and a client process is performed for operating another object written in a second

object-oriented language. The object operated in the server process is utilized by the client process by performing a communication process through a communications path between the server process and the client process. The server process operates a server class written in the first object-oriented language and operates a communications manager for managing communications with the client process. The client process operates a virtual server class for defining a method corresponding to a method defined in the server class to utilize the method defined in the server class. The client process also operates an object correspondence table for storing correspondence information between the object operated in the server process and the object operated in the client process, and operates a communications manager for managing the communications with the server process.

Current US Original Classification (1):
709/315

Full	Title	Citation	Front	Review	Classification	Date	Reference	Sequences	Attachments
------	-------	----------	-------	--------	----------------	------	-----------	-----------	-------------

KWIC	Draw Desc	Image
------	-----------	-------

[Generate Collection](#)[Print](#)

Term	Documents
VIRTUAL	55916
VIRTUALS	3
SERVER\$	0
SERVER	36819
SERVERA	9
SERVERABILITY	1
SERVERABLE	11
SERVERABLY	2
SERVERACCEPT	1
SERVERACCESS	1
"SERVERACCESS.JAVA"	1
(L1 AND (VIRTUAL ADJ1 SERVER\$).AB.).USPT.	5

[There are more results than shown above. Click here to view the entire set.](#)

Display Format:

[KWIC](#)[Change Format](#)[Previous Page](#)[Next Page](#)